

---

# Sistema de fitxers amb control de versions automàtic

---

26 d'Octubre de 2017

*Demian Cozzi Pusso*  
Grau en Enginyeria Informàtica  
Especialitat en Tecnologies de la Informació

Amb el suport i la direcció de:  
*Juan José Costa Prats*  
Departament d'Arquitectura de Computadors

## Sumari

<b>Resum</b>	<b>3</b>
<b>Resumen</b>	<b>4</b>
<b>Abstract</b>	<b>5</b>
<b>Capítol 1. Introducció</b>	<b>7</b>
1.1 Context	7
1.2 Abast del projecte	8
1.2 Motivació	8
1.3 Els actors implicats	9
1.4. Definició d'objectius	9
<b>Capítol 2. Estat de l'art</b>	<b>11</b>
2.1 Sistema de fitxers Files-11	11
2.2 Sistema controlador de versions Git	12
2.3 Sistema de fitxers ZFS	13
<b>Capítol 3. Metodologia i rigor</b>	<b>15</b>
3.1 Mètode de treball	15
3.2 Eines de seguiment	15
3.3 Mètode de validació	16
<b>Capítol 4. Planificació temporal</b>	<b>17</b>
4.1 Definició de tasques	17
4.2 Recursos humans	19
4.3 Recursos materials	20
4.4 Valoració d'alternatives i pla d'acció	21
<b>Capítol 5. Gestió econòmica</b>	<b>23</b>
5.1 Costos directes	23
5.2 Costos indirectes	24
5.3 Contingències	24
5.4 Imprevistos	25
5.5 Impostos	25
5.6 Cost total	26
5.7 Cost de les desviacions	26
<b>Capítol 6. Informe de sostenibilitat</b>	<b>27</b>
6.1 Estudi de l'impacte ambiental	28
6.2 Estudi de l'impacte econòmic	29
6.3 Estudi de l'impacte social	29
6.4 Matriu de sostenibilitat	30
6.5 Conclusions	30

<b>Capítol 7. Especificació</b>	<b>31</b>
7.1 Visió general de la solució	31
7.2 Base del sistema	32
7.3 Versionat de fitxers	32
7.3 Connexió a un servidor remot	33
<b>Capítol 8. Disseny</b>	<b>35</b>
8.1 Arquitectura del sistema	35
8.2 Base del sistema	36
8.3 Versionat de fitxers	37
8.4 Connexió a un servidor remot	39
<b>Capítol 9. Implementació</b>	<b>41</b>
9.1 File System in User Space - FUSE	41
9.2 Versionat de fitxers	42
<b>Capítol 10. Anàlisi de Resultats</b>	<b>45</b>
<b>Capítol 11. Conclusions</b>	<b>47</b>
<b>Justificació de les competències tècniques</b>	<b>48</b>
<b>Referències</b>	<b>49</b>

## Resum

En aquest document es reflecteix el treball realitzat per a un projecte destinat a donar forma a un sistema de fitxers que permet la gestió no només de la informació emmagatzemada actualment en aquest, sinó també de la capacitat d'administrar versions antigues dels fitxers.

Durant els primers capítols es pot observar el treball previ realitzat per dur a terme l'execució del projecte, aprofundint en cada aspecte que s'ha de tenir en compte per complir amb els objectius proposats.

En els següents capítols s'adjunta la documentació tècnica de la solució proposada, composta per l'especificació, el disseny del sistema, la seva implementació, i l'anàlisi de resultats obtinguts.

Al final del document es troben les conclusions que es poden extreure una vegada acabat el projecte, i la justificació de les competències tècniques assolides.

## Resumen

En este documento se refleja el trabajo realizado para un proyecto destinado a dar forma a un sistema de ficheros que permite la gestión no solo de la información almacenada actualmente en él, sino también de la capacidad de administrar versiones antiguas de los ficheros.

Durante los primeros capítulos se puede observar el trabajo previo realizado para llevar a cabo la ejecución del proyecto, profundizando en cada aspecto que se debe tener en cuenta para cumplir con los objetivos propuestos.

En los siguientes capítulos se adjunta la documentación técnica de la solución propuesta, compuesta por la especificación, el diseño del sistema, su implementación y el análisis de los resultados obtenidos.

Al final del documento se encuentran las conclusiones que pueden extraerse una vez terminado el proyecto, y la justificación de las competencias técnicas que se han alcanzado.

## Abstract

This document reflects the work done for a project that gives shape to a file system to let the management not only of the information currently stored in it, but also the ability to administer old versions of files.

During the first chapters, there is a previous work done in order to proceed to the execution of the project, deepening in every aspect that must be kept into account in order to satisfy the proposed objectives.

The following chapters have attached the technical documentation for the proposed solution, which is composed of the specification, the system design, its implementation, and the analysis of the obtained results.

At the end of the document there are found the conclusions that can be extracted once the project is finished, and a justification of the achieved technical skills.



# Capítol 1. Introducció

En aquest capítol s'introdueix el projecte en el seu context i l'abast que tindrà. A més, es justifica la motivació que justifica la seva realització i es fa un estudi dels possibles actors implicats en la solució. Al final del capítol s'estableixen els objectius que cal tenir presents en aquest projecte.

## 1.1 Context

Sense un sistema de fitxers, les dades emmagatzemades es veurien com una sèrie de bits, els quals resulten una forma molt difícil de manegar la informació. És per això que cal d'un suport software que arbitri i identifiqui el significat d'aquests bits per tal que l'usuari final treballi a un nivell d'abstracció que li permeti realitzar les seves activitats.

Els sistemes de fitxers són un subsistema dels sistemes operatius que s'encarreguen de controlar com s'emmagatzemen i s'extreuen les dades ubicades als dispositius d'emmagatzemament. Per la seva naturalesa, l'usuari es pot veure en la necessitat de tornar a una versió anterior d'un fitxer que hagi modificat. Això és un problema que la majoria de sistemes que els usuaris utilitzen majoritàriament no tenen en compte però que idealment caldria resoldre.

La finalitat que es vol aconseguir es que l'usuari tingui constància de que qualsevol canvi que faci en les seves dades queda registrat i pugui ser consultable i, a més, es pugui recuperar qualsevol estat en què es trobaven.

Hi ha solucions que ja atenen a aquesta necessitat (en el capítol 2 s'analitza un sistema de fitxers amb amb versionat anomenat Files-11), però manquen en altres aspectes que actualment s'estan abordant amb més diligència, com ara l'encryptació de les dades, la fiabilitat d'aquestes, o encara més eficiència en la seva administració i accés.



## 1.2 Abast del projecte

En l'abast, es tracta de posar uns límits a la solució proposada per tal de que el projecte sigui factible en el temps disponible.

S'intentaran utilitzar totes les eines i les tecnologies disponibles a l'abast per tal de satisfer els objectius establerts. D'aquesta manera es podrà reduir significativament el temps requerit per realitzar un projecte com aquest, que depenent del grau d'immersió, pot arribar a ser molt complexe.

Els únics obstacles que apareixeran de manera natural són propis d'un projecte educatiu com aquest. És a dir, s'espera que a mesura que es realitza el projecte s'aprenguin noves eines i tecnologies i es posin en pràctica.

No es pretén desenvolupar un sistema de fitxers des de zero. Això seria inviable per a un projecte de final de grau d'on penja un sol individu. El que es pretén és veure els resultats que donaria una solució complexa a partir d'una prova de concepte que prescindeix de les complicacions que no resulten primordials abordar, majoritàriament perquè depenen de tecnologies sobre la qual apareixen però que es troben fora del nostre abast.

## 1.2 Motivació

La motivació del projecte és clau per emprendre en la seva execució. Sense una raó que doni resposta a per què es fa una solució, no tindria sentit dedicar l'esforç que un projecte requereix. Per una banda, hi ha la motivació principal, que està lligada amb la definició del problema ja establert quan es va parlar del context del projecte. Per altra, hi ha la motivació personal.

La principal motivació es donar una solució a un problema existent per a l'usuari comú. L'usuari comú d'un sistema operatiu encara pateix pèrdues de dades. Però no sempre és degut a un error del sistema software o un problema físic en els dispositius d'emmagatzematge. Moltes vegades, l'usuari es pot trobar en la situació de voler tornar a un estat anterior del que ha fet. El problema, però, és que no es troba en la majoria de sistemes que hi ha actualment. El que permetria el sistema que es vol desenvolupar es pot dir que es com tornar enrere en el temps, però només amb l'estat dels fitxers que l'usuari té emmagatzemats en un ordinador, permetent a l'usuari satisfer aquesta necessitat.

A més, el projecte tindrà un caràcter educatiu. Aquest projecte és un treball de final de Grau en enginyeria informàtica. Com a projecte final d'uns estudis d'enginyeria s'imposa un repte personal que anima a l'estudiant al desenvolupament d'un sistema que tingui la capacitat d'aportar millor qualitat de vida a les persones allà on sigui possible, per mínim que sigui.

### 1.3 Els actors implicats

En aquesta secció es tracta de veure quins són els possibles actors a qui va destinada la solució proposada en aquest projecte. Qui o què pot utilitzar un sistema de fitxers? En distingim dos tipus d'actors: Els usuaris i el software client.

Respecte als usuaris, es fa referència als usuaris finals dels sistemes operatius que volen gestionar els seus fitxers. Per fer-ho utilitzaran software que proveeix accions directes a les seves dades. Tal software pot ser tant les comandes que permeten realitzar aquestes accions a través d'un terminal, com el software gràfic més comunament utilitzat per la major part d'usuaris, genèricament anomenat *explorador de fitxers*. Aquests usuaris, poden exercir qualsevol professió a l'hora d'utilitzar-lo, però cal tenir en compte que pot no tenir cap tipus de coneixement d'ordinadors.

Per una altra banda, existeix el software client que accedeix al sistema de fitxers per gestionar dades que no són directament utilitzades per l'usuari, però per raons exclusives de cada software necessiten accedir als sistemes d'emmagatzematge. En termes generals, parlem de fitxers relacionats amb la configuració que el software client emmagatzema (per aportar flexibilitat a cada usuari), o de fitxers temporals que el software client utilitzi durant la seva execució.

### 1.4. Definició d'objectius

L'objectiu principal d'aquest projecte és **desenvolupar un sistema de fitxers amb control de versions automàtic per als sistemes operatius Linux**.

El sistema amb versionat automàtic permetrà a un usuari comú gestionar els seus fitxers segons l'estat en el qual es trobaven anteriorment. A més, el sistema podrà realitzar còpies de seguretat dels fitxers a sistemes remots.

S'espera, doncs, que el sistema a desenvolupar tingui les següents característiques:

- **Ser un sistema de fitxers funcional.** Haurà de tenir les funcionalitats mínimes que s'esperen d'un sistema de fitxers, tals com crear directoris, emmagatzemar fitxers, o moure fitxers a un altre directori.
- **Versionar fitxers de forma automàtica.** El sistema de fitxers ha de versionar de forma automàtica els fitxers que un usuari i/o el seus programes modifiquen.
- **Permetre la gestió de versions.** El sistema de fitxers ha de permetre a l'usuari gestionar les versions dels seus fitxers.
- **Permetre realitzar còpies de seguretat.** El sistema permetrà transferir fitxers versionats a un servidor remot, per tal de que l'usuari disposi de còpies de seguretat de les versions dels fitxers que li interessi conservar.



## Capítol 2. Estat de l'art

En aquest capítol s'introdueixen alguns sistemes compleixen els objectius establerts, o parts d'aquests objectius. Fer un estudi de solucions similars facilita donar forma a la solució que es vol realitzar, i ajuda a veure si la solució proposada pot encara satisfer necessitats que els actors implicats no tinguin ateses.

D'entre totes les múltiples solucions existents en el mercat, s'han escollit els tres sistemes següents per analitzar: Files-11, Git, i ZFS. A continuació es fa una petita descripció de cadascun d'ells. Només es tractarà de donar una visió enfocada a les funcionalitats que tenen més pes per al projecte a desenvolupar.

### 2.1 Sistema de fitxers Files-11

Files-11<sup>1</sup> és un sistema de fitxers jeràrquic, on un fitxer pot ser fill de més d'un directori.

El versionat de fitxers realitzat en aquest sistema és molt bàsic: cada cop que es guarda un fitxer es crea una versió d'aquest, generant un nou fitxer que té el mateix nom que l'original seguit de la cadena ";X", on X es un nombre natural. Si només hi ha una versió del fitxer, per defecte X és 1. A mesura que es fan noves versions, es van generant còpies d'aquest on X varia de manera incremental.

La ubicació d'un fitxer es realitza mitjançant una identificació única amb el següent aspecte:

```
NODE"accountname password":device:[directory.subdirectory]filename.type;ver
```

i amb el següent significat:

- **NODE**: és una identificació a nivell de xarxa.
- **accountname**: Identificació d'usuari.
- **password**: Password de l'usuari.
- **device**: Dispositiu d'emmagatzemament.
- **directory**: directori arrel on s'ubica el fitxer o un subdirectori del fitxer. Pot no existir.
- **subdirectory**: Subdirectori o llista de subdirectoris en ordre descendent on s'ubica el fitxer. Pot no existir.
- **filename**: Nom del fitxer.
- **type**: tipus del fitxer. En alguns sistemes també forma part del nom del fitxer.
- **ver**: versió del fitxer.

El versionat es realitza de manera automàtica: Cada vegada que es tanca un fitxer, que en Linux equival a la crida de sistema *close*, es genera una nova versió com s'ha explicat abans. Tot i que aconseguir complir amb els objectius proposats, aquest procés pot arribar a omplir el directori de l'usuari, fet que dificultaria la gestió dels seus fitxers.

---

<sup>1</sup> "Guide to OpenVMS File Applications - Software Products Library."  
[http://h30266.www3.hp.com/odl/i64os/opsys/vmsos84/documentation/pdf/OVMS\\_731\\_file\\_app.PDF](http://h30266.www3.hp.com/odl/i64os/opsys/vmsos84/documentation/pdf/OVMS_731_file_app.PDF).  
Accessed 28 Feb. 2017.

## 2.2 Sistema controlador de versions Git

Git<sup>2</sup> no és un sistema de fitxers sinó un sistema de control de versions pensat especialment per a projectes de software on molta gent col·labora editant els mateixos fitxers, fins i tot quan s'editen al mateix moment. Per tant, disposa d'un gran ventall de funcionalitats, algunes de les quals poden servir d'inspiració per al nostre projecte.

El versionat en Git depèn de l'estat, en conjunt, dels fitxers pertanyents a un projecte de software. Les diferents versions queden emmagatzemades en el repositori del projecte.

Quan es treballa amb un estat del projecte en concret, l'usuari pot decidir arbitràriament, per cada fitxer, mantenir les modificacions realitzades fins l'últim moment en una àrea que emmagatzema els canvis que encara no es consideren permanents. Després pot confirmar els canvis fets a alguns dels seus fitxers, que passen a generar un nou estat del projecte en el repositori per a què estigui a l'abast de l'usuari. Algunes de les comandes que permeten aquest procés son:

\$ git status
Llista els fitxers nous o modificats per ser afegits al nou estat en el repositori.
\$ git add [fitxer]
Emmagatzema una còpia del fitxer, sense afegir-lo al nou estat. Si no es s'explicita cap fitxer, emmagatzemen tots els canvis.
\$ git reset [fitxer]
Elimina la còpia del fitxer emmagatzemada.
git commit
Confirma els canvis emmagatzemats creant un nou estat en el repositori.

A banda d'això, Git proporciona un model de gestió de branques, destinades principalment a gestionar canvis entre múltiples usuaris, o canvis que comporten funcionalitats addicionals.

En contrast amb Files-11, Git permet a l'usuari conservar les versions dels seus fitxers arbitràriament, i manté un directori ".git", que pot ser visible per l'usuari, on s'emmagatzemen tots els canvis fets al projecte amb aquest sistema.

---

<sup>2</sup> "Git - Git Basics." <https://git-scm.com/book/en/Getting-Started-Git-Basics>. Accessed 28 Feb. 2017.

## 2.3 Sistema de fitxers ZFS

El sistema de fitxers ZFS<sup>3</sup> és també alhora un sistema gestor de volums. Igual que Git, ZFS utilitza snapshots per emmagatzemar els canvis dels fitxers.

En quant a la gestió de volums, ZFS administra els dispositius d'emmagatzemament de manera que crea una agrupació d'aquests anomenada *storage pool*. La creació dels sistemes de fitxers es fa sobre aquesta agrupació que es veu com una únic volum d'emmagatzematge. Això permet una major flexibilitat a l'hora de gestionar l'emmagatzemament, doncs ja no es depèn de múltiples volums que s'han de configurar un a un. A més, permet afegir nous dispositius a l'agrupació per ampliar l'espai disponible sense cap configuració. Aquesta característica d'encarregar-se de totes les unitats d'emmagatzemament també permet a ZFS un millor control de la redundància i una millor capacitat per reparar les dades corruptes.

ZFS treballa a nivell de blocs de dades. Quan un bloc es modifica es crea una còpia nova independent de l'anterior. Qualsevol versió d'un fitxer encara es pot recuperar ja que tots els seus blocs tant presents com passats continuen existint.

---

<sup>3</sup> "What Is ZFS? - Oracle Solaris ZFS Administration Guide."  
[https://docs.oracle.com/cd/E23823\\_01/html/819-5461/zfsover-2.html](https://docs.oracle.com/cd/E23823_01/html/819-5461/zfsover-2.html). Accessed 28 Feb. 2017.



## Capítol 3. Metodologia i rigor

En aquest capítol es descriuen el mètode de treball que s'ha aplicat en l'execució del projecte, les eines de seguiment per garantir que es segueix aquest mètode, i el mètode de validació per verificar que la solució és correcta.

### 3.1 Mètode de treball

D'entre tots els mètodes de treball disponibles per a un projecte de software, s'aplicarà el mètode de desenvolupament en cascada. Amb aquest mètode el projecte es divideix en unes fases ben diferenciades que es desenvoluparan successivament.

Les fases que s'han executat en aquest ordre, són Especificació, Disseny, Implementació, Verificació i Refinament. A continuació es fa una breu explicació d'aquestes fases:

1. **Especificació:** Durant aquesta fase es fa un estudi del context, identificant les parts interessades, així com els requisits que haurà de satisfer el sistema.
2. **Disseny:** A partir de la fase d'especificació, es realitzarà el model del sistema tenint en compte la tecnologia que s'utilitzarà.
3. **Implentació:** A partir del disseny del sistema, es programa el codi que donarà forma al sistema empleant les eines i tecnologies a l'abast dels desenvolupadors.
4. **Verificació:** Durant aquesta fase es realitzaran una serie de proves per tal de verificar que el sistema funciona correctament.
5. **Refinament:** Aquesta fase addicional serveix per afegir noves funcionalitats que no hagin estat contemplades durant el marge de desenvolupament. Conté les fases anteriors de manera addicional, però en un marge de temps més petit.

### 3.2 Eines de seguiment

Per tal de planificar i executar les diferents fases del projecte s'ha realitzat un diagrama de Gantt. En aquest diagrama s'han marcat temporalment totes les fites establertes per reunions, entrega de documents i seguiment de les diferents fases.

També s'ha utilitzat el sistema de correu per tal de mantenir contacte mutu entre les persones implicades en el projecte per a que puguin acordar les dates de reunions, assabentar-se de canvis en la planificació, i altres necessitats que sorgeixin.



### 3.3 Mètode de validació

El mètode de validació empleat durant les primeres fases d'especificació i disseny serà la revisió mútua de les característiques que es vol que el sistema obtingui. Per fer-ho es contrastaran diferents punts de vista, especialment entre alumne i director.

Per assegurar el correcte funcionament del treball fet a la fase d'implementació, es realitzen jocs de proves per confirmar que les funcionalitats que es van implementant funcionin correctament. Aquests jocs de proves corresponen a la posterior fase de verificació, que permet assegurar que el funcionament final del sistema és com s'espera.

Durant la fase de refinament, les fases d'especificació, disseny i implementació passen per el mateix mètode de validació, que s'executen en un marge de temps més acotat.

## Capítol 4. Planificació temporal

En aquest capítol es planifica el projecte: es defineixen les tasques que calguin i s'analitzen i es gestionen els recursos disponibles. També dedica un moment a realitzar una valoració d'alternatives i el pla d'acció a realitzar en cas de que sorgeixin incidents en el pla establert. Al final del capítol es troba el diagrama de Gantt a pàgina completa, per una millor visibilitat.

### 4.1 Definició de tasques

Dividim la planificació del projecte en 5 fases generals: Especificació, Disseny, Implementació, Verificació i Refinament. Aquestes fases ja van ser establertes per el mètode de treball que s'aplicarà al projecte, però cal fer una adaptació que expliqui cada component amb més detall per poder realitzar aquesta solució en particular.

A la següent pàgina es troben planificades les fases del projecte amb les tasques corresponents a cadascuna. Hi ha una fase addicional anomenada Gestió del projecte (Fase 0). En aquesta fase s'apliquen els coneixements impartits per l'assignatura *Gestió de Projectes*.

Al final del capítol es pot observar el diagrama de Gantt utilitzat per dividir el temps en les tasques descrites anteriorment.

<b>FASE 0. [5 setmanes] GESTIÓ DEL PROJECTE</b>
<b>FASE 1. [5 setmanes] ESPECIFICACIÓ</b>
<b>1.1 [2 setmanes] Definició del projecte</b> S'inicien les tasques oportunes per tal de passar a la següent fase (i.e., parlar amb el director, aclarir l'objectiu global,...).
<b>1.2 [3 setmanes] Anàlisi de requisits</b> S'investiguen, s'analitzen i s'estableixen els requisits que caldran, i es donarà forma al document amb l'especificació del sistema.
<b>FASE 2. [2 setmanes] DISSENY</b>
<b>2.1 [2 setmanes] Sistema de fitxers amb versionat</b> Fase on es dissenya el sistema de fitxers amb versionat.
<b>2.2 [2 setmanes] Connexió amb servidor remot</b> Fase on es dissenya la part del sistema que connecta amb el servidor remot.
<b>FASE 3. [3 setmanes] IMPLEMENTACIÓ</b>
<b>3.1 [3 setmanes] Implementació del sistema de fitxers amb versionat</b> S'implementa el sistema de fitxers amb versionat.
<b>3.2 [3 setmanes] Implementació de la connexió amb servidor remot</b> S'implementa la part del sistema que connecta amb el servidor remot.
<b>FASE 4. [1 setmana] VERIFICACIÓ</b>
<b>4.1 [1 setmana] Verificació del sistema de fitxers amb versionat</b> Fase on es verifica el sistema de fitxers amb versionat.
<b>4.2 [1 setmana] Verificació de la connexió amb servidor remot</b> Fase on es verifica la part del sistema que connecta amb el servidor remot.
<b>FASE 5. [4 setmanes] REFINAMENT</b>
<b>5.1 [1 setmana] Anàlisi de nous requisits</b> Es dona forma a requisits nous requisits que hagin sorgit i/o que no s'hagin tingut en compte durant la fase d'especificació.
<b>5.2 [1 setmana] Disseny de noves funcionalitats</b> Es dissenyen noves funcionalitats del sistema que satisfacin els nous requisits.
<b>5.3 [1 setmana] Implementació de noves funcionalitats</b> Es programen les noves funcionalitats.
<b>5.4 [1 setmana] Verificació</b> Es verifiquen les noves funcionalitats.
<b>Temps total estimat: 15 Setmanes</b>

## 4.2 Recursos humans

A continuació s'anomena i es descriu el personal del projecte:

<b>Cap de projecte</b>
Persona encarregada de realitzar la planificació del projecte, executar les fases pertinents i controlar les diferents activitats.
<b>Analista</b>
S'encarrega d'analitzar el context, les parts interessades i els seus objectius, i l'abast del projecte per tal de dotar a la solució de les característiques necessàries per satisfer els requisits del projecte.
<b>Dissenyador</b>
Dissenya l'arquitectura de la solució a partir de models conceptuals.
<b>Programador</b>
Implementa la solució utilitzant les tecnologies disponibles al seu abast.
<b>Tester</b>
S'assegura que l'implementació de la solució és correcta utilitzant programes de testing amb els jocs de proves corresponents.

### 4.3 Recursos materials

Aquests són els recursos materials que el personal del projecte utilitzarà durant el seu desenvolupament. Cada recurs, a més, està assignat al personal que pertorqui:

<b>Ordinador Personal</b>
Màquina capaç d'executar operacions aritmeticològiques per al processament d'informació. Requereix d'un sistema operatiu per realitzar les tasques del projecte.
Assignacions: Cap de projecte, Analista, Dissenyador, Programador, Tester.

<b>Ubuntu</b>
Sistema operatiu que proveeix un ampli ventall de solucions software de lliure distribució que permetran el desenvolupament del projecte.
Assignacions: Cap de projecte, Analista, Dissenyador, Programador, Tester.

<b>Google Docs</b>
Processador de textos en línia.
Assignacions: Cap de projecte, Analista, Dissenyador.

<b>Ganttter</b>
Processador de diagrames de Gantt en línia.
Assignacions: Cap de projecte

#### 4.4 Valoració d'alternatives i pla d'acció

Encara que s'ha tingut la màxima cura a l'hora de determinar els temps, la naturalesa del projecte (de caire educatiu) comporta major incertesa a l'hora d'esbrinar el temps real que es requerirà en el seu desenvolupament. És per això que s'ha afegit un marge considerable anomenat "refinament" per tal de que, en cas de no complir amb els temps previstos en les fases crítiques anteriors, es pugui prescindir d'aquesta fase complementaria.

Per tant, la desviació només prevista durant la fase d'implementació, es considera manegable amb el temps addicional que es pot utilitzar en la fase de "refinament", i es confia en poder acabar el projecte en el temps establert.

Existeix també, però, el risc inherent a la dependència del projecte d'una o més persones. En aquest cas, com que el desenvolupament del projecte depèn únicament de la persona que realitza el treball, aquest risc s'ha de tenir present.

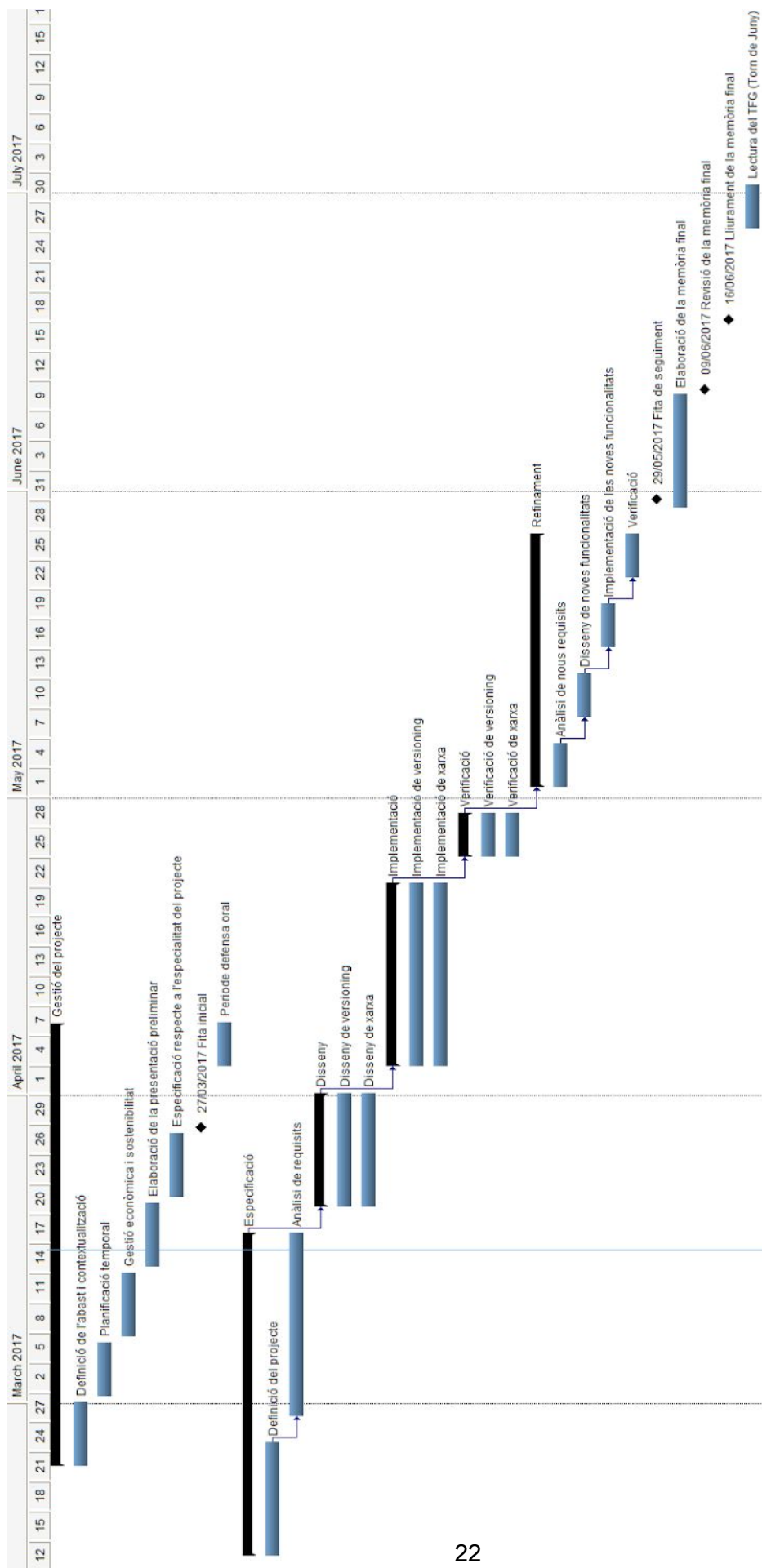


Figura 1. Diagrama de Gantt del projecte.

## Capítol 5. Gestió econòmica

En aquest capítol s'estimen els costos del projecte. Abans però, s'identifiquen alguns dels costos per tal d'associar-los a certes activitats del projecte. Això permetrà disposar d'una major precisió en l'estimació del cost total.

En diferenciem entre costos directes i costos indirectes. Els primers son aquells costos directament atribuïbles a les activitats del projecte. Els segons, son aquells que no ho son. Dins d'un projecte de software els costos materials tenen associats un costos en recursos hardware i software.

### 5.1 Costos directes

El costos directes son aquells costos directament atribuïbles a les activitats del projecte, les quals es troben especificades a la secció *Planificació Temporal*. En diferenciem entre els costos en recursos humans i els costos en recursos materials.

Els costos en recursos humans es comptabilitzen per hores invertides en el projecte, i per cada rol existent en el projecte s'assigna un salari. El salari de cadascun dels rols establerts en la secció 3.1.3 (Recursos) és de 10 euros per hora. La següent taula calcula el cost total en recursos humans tenint en compte una dedicació al projecte de 8 hores per dia (Figura 2).

Tasca	Dies	Recurs humà	Cost (€)
Gestió del projecte	28	Cap de projecte	2240
Especificació	25	Analista	2000
Disseny	10	Dissenyador	800
Implementació	15	Programador	1200
Verificació	5	Tester	400
Refinament			
Anàlisi de nous requisits	5	Analista	400
Disseny de noves funcionalitats	5	Dissenyador	400
Implementació de les noves funcionalitats	5	Programador	400
Verificació	5	Tester	400
Cost Total Recursos Humans			6440

Figura 2. Cost en recursos humans



A continuació, s'analitza els costos en recursos materials:

- **Ordinador personal**

Degut a la manca d'informació respecte al preu d'adquisició del recurs i la variació dels components interns d'aquest equip en el temps, s'ha optat per donar un cost aproximat equivalent a una amortització de 40€ en la duració del projecte (aproximadament 120 dies), incloent l'IVA.

- **Eines de desenvolupament**

El software utilitzat és de llicència lliure, per tant, no es contempla cap cost.

## 5.2 Costos indirectes

Aquí es diferencien els costos que no són directament imputables en la seva execució. Només es contemplen despeses generals:

- **Consum elèctric de l'ordinador personal**

L'equip té un consum energètic de 300W. Considerant una jornada de 8 hores diàries i la duració del projecte (aprox. 120 dies), el cost s'aproxima als 44€.

- **Accés a Internet**

Amb una quota de 45€ mensuals i una jornada de 8 hores diàries per a un total de 120 dies, el cost aproximat és de 208€.

## 5.3 Contingències

Per tal d'atenuar els errors que puguin esdevenir, s'assumeix un cost addicional calculat respecte al valor total del pressupost. Aquest percentatge és mínim, ja que la naturalesa d'aquest projecte ja preveu que no n'hi haurà pràcticament despeses. Per tant, s'assumeix un cost addicional per contingències del 5%.

## 5.4 Imprevistos

Per als imprevistos, es calcula el cost que suposaria cada risc computant només la part equivalent a la probabilitat de que succeeixi tal imprevist.

**Risc 1:** Durant la fase d'implementació es triga més del previst.

- Probabilitat: 40%
- Actuació: Dedicar hores addicionals en la fase d'implementació.
- Temps màxim estimat: 20 dies.
- Cost associat: 1600€
- Cost computat: 640€

**Risc 2:** El sistema operatiu no carrega o no funciona correctament.

- Probabilitat: 10%
- Actuació: Reparació del sistema operatiu.
- Temps màxim estimat: 1 dia
- Cost associat: 10€
- Cost computat: 1€

**Risc 3:** Una actualització en una eina de software impossibilita l'avanç.

- Actuació: Reparació de l'eina de software.
- Temps màxim estimat: 1-2 dies
- Cost associat: 20€
- Cost computat al 10%: 2€

## 5.5 Impostos

S'aplica l'Impost sobre el Valor Afegit (IVA). En el nostre cas, correspon al 21% del cost dels recursos materials del projecte.

Si el projecte estigués relacionat amb una activitat econòmica, també s'aplicaria l'Impost sobre Activitats Econòmiques (IAE). En aquest cas, com que en la realitat el projecte no té cap activitat econòmica, es pot descartar la seva aplicació.

Cal mencionar que no s'obtenen beneficis en el marge de realització d'aquest projecte, per tant no s'aplica l'Impost de Societats (IS). Cal també mencionar que l'Impost sobre la Renda de les Persones Físiques (IRPF) no es contempla ja que recau directament sobre el personal una vegada ja han obtingut els beneficis corresponents a la seva activitat professional.

## 5.6 Cost total

La següent taula calcula el cost que suposaria l'execució d'aquest projecte (Figura 3). El pressupost necessari s'aproxima als 8.000 euros.

Concepte	Cost (€)
Costos directes	----
Costos en recursos humans	6440
Costos en recursos hardware	40
Costos en recursos software	0
Costos indirectes	----
Despeses generals	252
Imprevistos	643
Cost total (sense contingències)	7375
Contingències (5%)	369
<b>Cost total en brut</b>	<b>7744</b>

Figura 3. Cost aproximat del projecte abans de la seva execució.

## 5.7 Cost de les desviacions

Durant la fase d'implementació s'ha trigat més del previst. Es va assignar una durada addicional d'aproximadament 5 setmanes. Això comporta haver de calcular els costos addicionals deguts al risc associat amb aquest imprevist. En el nostre cas, el risc associat a aquest imprevist té un cost associat de 400€ per setmana. A més, les fases de verificació i refinament també s'han vist afectades, quedant aquestes relegades per el temps addicional a la fase d'implementació. El cost corresponent a aquestes fases és de 2000€.

El cost total del projecte continua sent el mateix, doncs el cost associat de les fases anteriors i el de les setmanes addicionals a la fase d'implementació és el mateix, i el temps addicional dedicat al projecte ha esdevingut del temps que no es va dedicar durant les primeres setmanes d'implementació i que no van quedar computades.

## Capítol 6. Informe de sostenibilitat

Aquest capítol inclou l'aplicació d'unes recomanacions suggerides que un enginyer suposadament ha de dur a terme durant la seva vida professional. Aquestes recomanacions tenen com a finalitat que l'estudiant realitzi un anàlisi de sostenibilitat del seu projecte.

En aquest informe de sostenibilitat, s'analitzen tres aspectes relacionats amb el projecte: La primera coneguda com *Projecte Posat en Producció (PPP)*, que inclou la planificació, el desenvolupament i la implantació del projecte, *la Vida Útil del projecte*, que comença una vegada aquest és implantat i acaba amb el seu desmantellament, i els riscos inherents al propi projecte durant tota la seva construcció i la seva vida útil.

Per cadascuna d'aquestes parts abans esmenades, s'estudia el seu impacte des de tres punts de vista: L'impacte ambiental, l'econòmic, i l'impacte social. A continuació s'analitzaran aquests aspectes tenint en compte cada punt de vista. Al final del capítol es fa un valoració del projecte calculant numèricament la seva sostenibilitat amb la *matriu de sostenibilitat*.

Els estudis que precisen de càlculs econòmics precisin en algun dels apartats es troben quantificats en el capítol dedicat a la gestió econòmica del projecte.

### 6.1 Estudi de l'impacte ambiental

L'estudi de l'impacte ambiental tracta d'estimar els recursos ambientals que consumirà el disseny del projecte.

#### Consum del disseny

En el capítol dedicat a la gestió econòmica es dedica una part a calcular el consum energètic que comporta realitzar aquest projecte. L'únic impacte ambiental que pot tenir el projecte és el del consum d'energia que té l'ús d'un ordinador personal, i que és el mètode més idoni per fer un projecte de software autònomament. Per tant, es pot considerar que els seus efectes durant l'elaboració del projecte són pocs.

Durant la fase de definició del projecte, s'ha tingut present l'ús de tecnologies i llibreries de software que facilitin el màxim possible el treball a fer, per tal de minimitzar el temps invertit, el qual es pot traduir en un menor ús dels recursos, que a la vegada es tradueix en un menor impacte ambiental.

Tot i això, cal dir que els recursos hardware utilitzats van ser manufacturats abans de l'any 2009, fet que comporta que els components que es van utilitzar en la seva fabricació molt probablement no tenien l'eficiència energètica dels components que hi ha manufacturats actualment.

## Petjada ecològica

Per una part, el disseny és modular. Aquesta característica que aporta flexibilitat a la solució comporta un sacrifici en l'eficiència de la seva implementació. A més, no s'espera tenir una implementació comparable a les implementacions comercials.

Fer un càlcul numèric en l'eficiència del software és inviable. Per saber la petjada ecològica que això comporta caldria fer un estudi comparatiu entre diferents implementacions utilitzant eines de mesura de rendiment dels programes (profiling). Això es troba fora de l'abast del projecte. Tot i això, es podria optar a fer una estimació, però comportaria un marge d'error molt alt.

## Riscos ambientals

Existeix un risc inherent a l'utilització de qualsevol recurs hardware que requereix d'electricitat. Tot i que les probabilitats són molt baixes, hi ha el risc de sofrir un incendi. Aquest risc pot una petjada ecològica, tot i que sigui despreciable.

## 6.2 Estudi de l'impacte econòmic

L'estudi de l'impacte econòmic planteja la viabilitat econòmica del projecte en tots els seus aspectes.

### Factura del projecte

En el capítol dedicat a la gestió econòmica es fa un anàlisi i una estimació dels costos que el projecte comporta.

S'ha de tenir en compte que el cost calculat té en compte el cost en recursos humans com el sou que rebrien els rols definits en la planificació temporal del projecte. Com que només hi ha una persona exercint tots els rols y a més és el cap de projecte, no es considera un exercici realista ja que no existeix cap cobrament.

### Pla de viabilitat

Hi ha sistemes de fitxers destinats a un software que l'usuari final ha de pagar per utilitzar-lo, i aquells on l'usuari final no paga res. El projecte pertany al segon tipus, per tant, no es contempla una viabilitat econòmica. A més, es desconeix quin impacte econòmic a nivell empresarial pot tenir el desenvolupament parcial d'un sistema operatiu que és el que s'està fent.

Aquesta solució pretén millorar experimentant amb unes funcionalitats que no han sigut històricament comunes en les solucions més populars. En cas de que tinguí un èxit considerable, sempre es pot tenir present la viabilitat econòmica, però només a l'hora de fer el seu desenvolupament gràcies als ajuts que les comunitats de software lliure reben per realitzar aquest tipus de solucions sense ànim de lucre.

Tot i això, com que no es contempla cap pla de viabilitat per la naturalesa del projecte, s'ha optat per la neutralitat en aquest aspecte.

## Riscs econòmics

No es conceben escenaris econòmics que hagin pogut perjudicar la viabilitat del projecte.

## 6.3 Estudi de l'impacte social

L'estudi de l'impacte social considera les implicacions socials especialment sobre el col·lectiu a qui va dirigit el projecte, però també a qualsevol col·lectiu que es pot veure afectat per aquest.

### Impacte personal

Com ja s'ha mencionat anteriorment, l'impacte personal que aquest projecte pot tenir és de caire educatiu: S'han obtingut coneixements respecte l'arquitectura de sistemes operatius, i en definitiva, ha comportat una experiència enriquidora en molts aspectes d'àmbit professional la realització d'aquest treball.

### Impacte social

Els sistemes operatius, i en efecte, els sistemes de fitxers, son molt presents en la vida de totes les persones que utilitzen els seus ordinadors. I mentre els sistemes operatius no aconsegueixin ser transparents als usuaris que volen utilitzar-los, sempre hi haurà millores a considerar. Això és precisament el que s'ha intentat millorar amb la solució d'aquest projecte, evitar que l'usuari hagi de tenir una preocupació més.

Tot i així, no es considera que aquest projecte tindrà un impacte social, per ser de caire únicament educatiu. La solució donada termina sent una solució més d'entre les múltiples solucions que intenten resoldre el mateix problema.

## Riscs socials

Degut al tipus d'implementació, hi ha un alt risc de que el sistema es trobi amb molts forats de seguretat, que comportarien l'exposició dels usuaris a atacs informàtics, quedant la seva privacitat compromesa. És per això, que per la naturalesa de caire educatiu del projecte, es recomana no utilitzar aquesta solució més enllà d'un entorn de proves.

## 6.4 Matriu de sostenibilitat

A continuació es mostra la matriu de sostenibilitat (Figura 4), que permet calcular, i per tant donar una valoració numèrica a la sostenibilitat del projecte.

	PPP	Vida útil	Riscs
Ambiental	Consum del disseny 9	Petjada ecològica 8	Riscs ambientals -1
Econòmic	Factura 10	Pla de viabilitat 10	Riscs econòmics 0
Social	Impacte personal 10	Impacte social 1	Riscs socials -15
Rang sostenibilitat	29	19	-16
	32		

Figura 4. Matriu de sostenibilitat

## 6.5 Conclusions

Com ja es pot veure, la sostenibilitat d'aquest projecte és gràcies, primordialment, a que tot el desenvolupament es realitza amb una computadora sense esdevenir costos molt grans, i sense tenir efectes ambientals. Cal tenir en compte però, que hi ha un risc social molt alt en el cas d'utilitzar aquesta solució a la pràctica degut a la dubtosa seguretat que comporta la implementació proposada.

## Capítol 7. Especificació

Aquí es troben llistades les característiques del sistema de fitxers. La cerca de característiques es fa mitjançant l'anàlisi dels requisits que l'usuari posseeix. Al disposar-nos a fer un sistema de fitxers, es parteix d'algunes característiques que ja s'espera que tingui un sistema de fitxers, i posteriorment afegirem les característiques corresponents al control de versions, i de connexió a un servidor remot.

### 7.1 Visió general de la solució

L'objectiu de la solució és permetre a l'usuari gestionar els seus fitxers a través de qualsevol variant de sistemes operatius GNU/Linux. El sistema ha de ser fàcil d'usar per qualsevol usuari sense necessitat de formació. El sistema ha de versionar automàticament els fitxers que siguin modificats, i també permetre a l'usuari la gestió d'aquests. El sistema també ha de poder realitzar còpies de seguretat dels fitxers versionats a una màquina remota.

A continuació es mostra una visió global del sistema a desenvolupar:

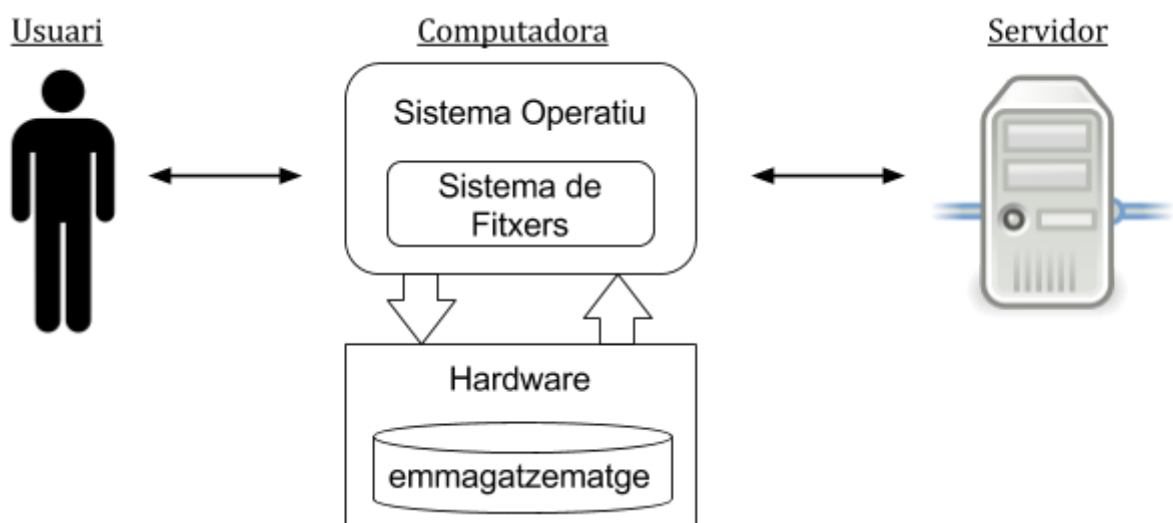


Figura 5. Visió general del sistema. El sistema de fitxers a desenvolupar s'integrarà en el sistema operatiu. El sistema de fitxers es comunica amb els sistemes d'emmagatzematge. També accedeix a un servidor remot.



## 7.2 Base del sistema

A continuació s'especifiquen els requisits funcionals del nostre sistema que donaran suport al funcionament bàsic. El sistema ha de permetre a l'usuari realitzar les tasques llistades:

1. Muntar el sistema de fitxers.
2. Desmuntar el sistema de fitxers.
3. Crear un fitxer.
4. Accedir a un fitxer.
5. Esborrar un fitxer.
6. Modificar el contingut d'un fitxer.
7. Crear un directori.
8. Accedir a un directori.
9. Esborrar un directori.
10. Moure un fitxer o directori.
11. Modificar el nom d'un fitxer o directori.
12. Consultar la data de modificació d'un fitxer o directori

## 7.3 Versionat de fitxers

L'usuari disposarà d'un sistema de versionat automàtic que versionarà un fitxer cada vegada que el seu contingut canviï, i l'emmagatzemarà a un espai dedicat a conservar aquestes versions. Aquest espai és invisible a ulls de l'usuari, i aquest podrà gestionar les versions a través d'un client.

Per limitacions de la memòria del sistema, no es pot emmagatzemar fitxers il·limitadament. Això comporta limitar l'espai dedicat al versionat automàtic, i disposar d'un servei que elimini versions antigues dels fitxers. Per evitar la pèrdua de versions que l'usuari vulgui conservar, s'afegeixen funcionalitats addicionals que permetin conservar els fitxers versionats.

Seguidament, es llisten les funcionalitats que l'usuari disposarà respecte al versionat de fitxers:

1. Consultar les versions disponibles d'un fitxer.
2. Restaurar una versió.
3. Conservar una versió indefinidament.
4. Etiquetar una versió.
5. Afegir un comentari a una versió.
6. Esborrar una versió.

### 7.3 Connexió a un servidor remot

L'usuari disposarà d'un servidor remot on poder emmagatzemar les versions que ha volgut conservar. Cada vegada que l'usuari vulgui conservar una versió indefinidament, el sistema li preguntarà si vol transmetre una còpia de seguretat. A continuació s'anomena la funcionalitat que l'usuari disposarà per dur a terme aquesta tasca:

1. Fer una còpia de seguretat d'una versió cap a un servidor remot.



## Capítol 8. Disseny

En aquest capítol queda plasmada l'arquitectura del sistema de fitxers que es vol dissenyar. El Kernel de Linux, el nucli del sistema operatiu sobre el qual el sistema de fitxers s'executa, cal tenir-lo present com a dependència tecnològica en el disseny.

### 8.1 Arquitectura del sistema

En el següent esquema es pot observar la visió general de l'arquitectura del sistema de fitxers en Linux. Aquest proveeix l'interfície *Virtual File System*<sup>4</sup> (VFS), una interfície que facilitar la labor als desenvolupadors d'aplicacions a l'hora d'accedir a les dades emmagatzemades ja que permet una gestió comuna independentment del sistema de fitxers que hi hagi al darrere. Això comporta que els desenvolupadors de sistemes de fitxers han de suportar aquesta interfície.

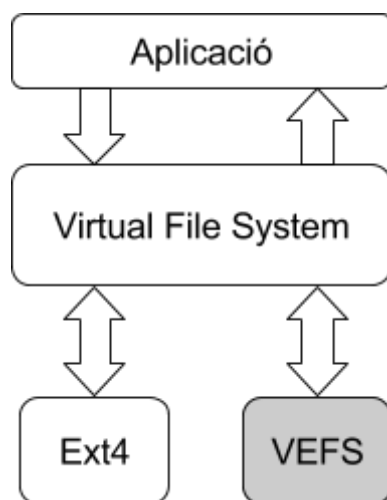


Figura 6. Arquitectura del sistema de fitxers en el Kernel de Linux. El sistema de fitxers VEFS que desenvolupem es comunica a través de la interfície especificada pel sistema operatiu. Ext4 és un altre sistema de fitxers, d'exemple per veure la flexibilitat que dona l'interfície *Virtual Filesystem*.

---

<sup>4</sup>Chapter 12 of "*Understanding the Linux Kernel*" by Daniel P. Bovet, Marco Cesati (O'Reilly Media, 2005), ISBN 0-596-00565-2

## 8.2 Base del sistema

La base del sistema és el punt de partida d'on penjaran les funcionalitats bàsiques. A continuació s'explica l'interfície que donarà suport al seu disseny.

L'interfície *Virtual File System*, requereix que els desenvolupadors de sistemes de fitxers la suportin per tal de donar una interfície comuna als desenvolupadors d'aplicacions.

Seguidament, es llisten les funcionalitats requerides del sistema VEFS que tenen relació amb les funcionalitats que especifica la interfície VFS. El sistema ha de ser compatible amb aquesta interfície.

VEFS	<i>Virtual File System</i>
Muntar el sistema de fitxers.	<i>mount</i>
Desmuntar el sistema de fitxers.	<i>umount</i>
Crear un fitxer.	<i>creat</i>
Accedir a un fitxer.	<i>open, read</i>
Esborrar un fitxer.	<i>unlink</i>
Modificar el contingut d'un fitxer.	<i>write, lseek</i>
Crear un directori.	<i>mkdir</i>
Accedir a un directori.	<i>readdir</i>
Esborrar un directori.	<i>rmdir</i>
Moure un fitxer o directori.	<i>rename</i>
Modificar el nom d'un fitxer o directori.	<i>rename</i>
Consultar la data de modificació d'un fitxer o directori	<i>stat</i>

En caldrà tenir presents altres funcionalitats que s'hauran d'implementar per satisfer la compatibilitat:

Funcionalitat	<i>Virtual File System</i>
Tancar un fitxer.	<i>close</i>
Sincronitzar l'estat del fitxer en memòria amb el dispositiu físic.	<i>fsync</i>

## 8.3 Versionat de fitxers

En aquest apartat s'expliquen els processos i les limitacions que donen lloc al sistema de versionat de fitxers.

Cada vegada que es procedeix a modificar el contingut d'un fitxer, el sistema de control de versions automàtic ha de generar un nou fitxer amb l'estat previ a la seva modificació. El nou fitxer resultant passa a ser una versió del fitxer modificat que s'emmagatzema a l'espai dedicat al versionat de fitxers.

El sistema ha d'emmagatzemar en dos llocs diferents els fitxers que l'usuari ha versionat dels fitxers versionats automàticament per fer més fàcil la tasca de neteja al servei de neteja del sistema de versionat automàtic. El lloc dedicat a l'emmagatzemament de versions que l'usuari ha decidit conservar no compta per al límit d'espai dedicat a les versions automàtiques. S'assumeix que l'usuari vol conservar aquestes versions, i per tant, es conserven com a espai d'usuari.

A l'espai dedicat al versionat de fitxers automàtic, cada fitxer disposa d'un directori propi ubicat al mateix directori del fitxer original, d'on pengen les seves versions. El nom d'aquest directori és el mateix que el del fitxer original. Aquesta forma facilita l'implementació del sistema de fitxers versionats.

El sistema ha d'identificar cada versió d'un fitxer amb una cadena de caràcters autogenerat mitjançant *hashing*, per tal de distingir entre diferents versions. Cada fitxer amb versions automàtiques disposarà d'un fitxer de metadades en el seu directori en l'espai dedicat al versionat automàtic de fitxers.

El fitxer amb metadades de les versions automàtiques contindrà:

- L'identificador de la versió més actual del fitxer versionat.

i per cada versió:

- L'identificador de la versió.
- La data de versionat.
- Un camp opcional per emmagatzemar una etiqueta que l'usuari vulgui afegir a la versió.
- Un camp opcional per emmagatzemar un comentari que l'usuari vulgui afegir a la versió.

A l'espai dedicat a emmagatzemar les versions que l'usuari ha volgut conservar, es segueix la mateixa estructura de directoris, però canvia el fitxer de metadades, que no tindrà cap identificador de la versió més actual.

A continuació es procedeix a realitzar les funcionalitats requerides. Les funcionalitats contenen els camps que reben de la petició (entre parèntesis).

Consultar les versions disponibles d'un fitxer (String *path*)

- 1) El sistema llista les versions automàtiques disponibles referents al fitxer ubicat a *path*.  
Per cada versió: es mostra l'identificador, la data de versionat, el tag, i el comentari.
- 2) El sistema llista les versions conservades manualment.

Restaurar una versió (String *path*, Integer *id*)

- 1) El sistema genera una nova versió automàtica referent al fitxer ubicat a *path*.
- 2) El sistema esborra el fitxer ubicat a *path*.
- 3) El sistema genera una còpia de la versió amb identificador *id* a l'ubicació *path*.

Conservar una versió indefinidament (String *path*, Integer *id*)

- 1) El sistema genera una versió manual amb identificador *id* referent a la versió del fitxer ubicat a *path*.

Etiquetar una versió (String *path*, Integer *id*, String *tag*)

- 1) El sistema emmagatzema la metadada *tag* per a la versió referent al fitxer ubicat a *path*, amb identificador *id*).

Afegir un comentari a una versió (String *path*, Integer *id*, String *Comentari*)

- 1) El sistema emmagatzema la metadada *Comentari* per a la versió amb identificador *id* referent al fitxer ubicat a *path*.

Esborrar una versió (String *path*, Integer *id*).

- 1) El sistema elimina la versió automatitzada i la versió manual amb identificador *id* referents al fitxer ubicat a *path*.

## 8.4 Connexió a un servidor remot

El sistema ha de permetre la còpia de versions que l'usuari ha volgut conservar cap a un servidor remot. El servidor ha d'executar un servei que espera rebre peticions del client.

A continuació es procedeix a realitzar les funcionalitats requerides. Les funcionalitats contenen els camps que reben de la petició (entre parèntesis).

Fer una còpia de seguretat d'una versió cap a un servidor remot (String *path*, String *ip*, Integer *port*, String *user*, String *password*).

- 1) El client crea una connexió amb el servidor amb a partir de la direcció IP *ip*, el port *port*, el nom d'usuari *user* i la contrasenya *password*.
- 2) Si la connexió ha sigut establerta, el client transmet les dades del fitxer ubicat a *path*.
- 3) El client demana el tancament de la connexió.





## Capítol 9. Implementació

Aquí es mostren els diferents mètodes i eines d'implementació que s'han utilitzat per tal de donar forma final al disseny proposat en el capítol anterior. L'eina que cal destacar que ha donat suport a l'implementació del sistema és la interfície FUSE.

### 9.1 File System in User Space - FUSE

Per tal de facilitar la implementació del sistema s'ha utilitzat la interfície FUSE<sup>5</sup> (Filesystem in Userspace) que permet implementar el nostre sistema de fitxers a nivell d'usuari. Això té alguns avantatges però també comporta alguns inconvenients.

Per una part, ens permet prescindir d'implementar la base del sistema per tal de centrar-se en les característiques que es volen afegir. Però tenir el sistema VEFS a nivell d'usuari sacrifica eficiència ja que comporta molts canvis de context entre nivell d'usuari i nivell de sistema. A més implica un menor grau de seguretat la dependència múltiple entre nivells, ja que trenca amb la separació de nivells.

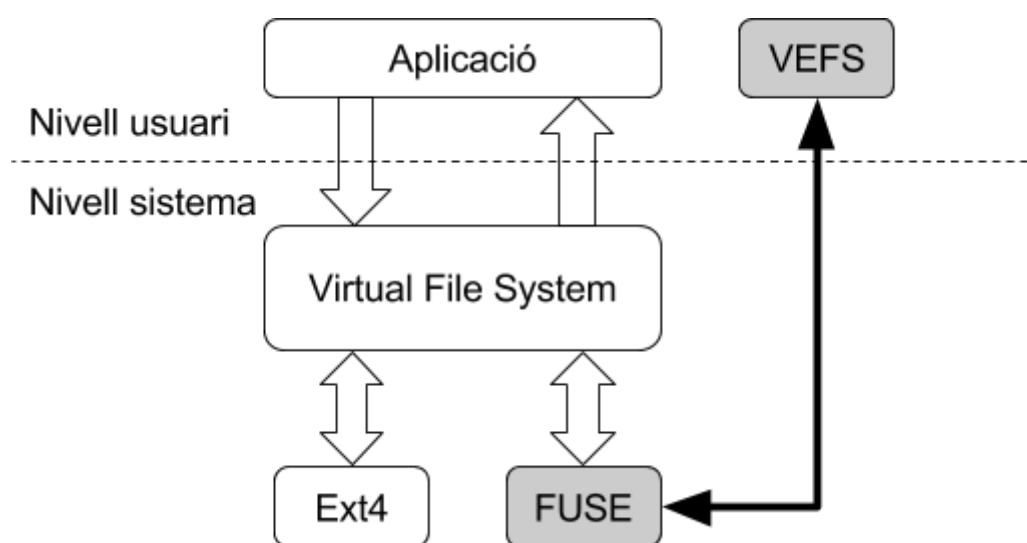


Figura 7. Interfície File System in User Space - FUSE. El sistema es comunica amb aquesta interfície.

<sup>5</sup> "Libfuse - GitHub." <https://github.com/libfuse/libfuse>. Accessed 19 Oct. 2017.

L'eina permet enllaçar les operacions del Virtual Filesystem de Linux amb operacions locals, que són les que acaben executant-se quan es crida a alguna de les primeres. A la Figura 8 es pot observar la definició de l'estructura que permet fer l'enllaç.

```
static struct fuse_operations vefs_oper = {
    .init      = vefs_init,
    .getattr   = vefs_getattr,
    .access    = vefs_access,
    .readdir   = vefs_readdir,
    .mknod     = vefs_mknod,
    .mkdir     = vefs_mkdir,
    .rmdir     = vefs_rmdir,
    .rename    = vefs_rename,
    .open      = vefs_open,
    .create    = vefs_create,
    .read      = vefs_read,
    .write     = vefs_write,
    .fsync     = vefs_fsync,
    .ioctl     = vefs_ioctl,
};
```

Figura 8. Estructura que enllaça les crides del Virtual Filesystem amb les operacions FUSE.

El sistema de fitxers implementat permetia que els directoris relacionats amb el sistema de versionat de fitxers puguin ser visibles a l'usuari. Es va haver de modificar la funció *readdir* per tal de que no apareguin cada vegada que l'usuari consulti el contingut del directori arrel del sistema de fitxers.

## 9.2 Versionat de fitxers

Per implementar les funcionalitats que l'usuari utilitza per gestionar els seus fitxers, s'ha aprofitat l'ús que es pot donar a les crides de la funció *ioctl* definida al Virtual Filesystem.

Aquesta funció tradicionalment s'utilitza per transmetre comandes a dispositius. En el nostre cas, com que FUSE té enllaçada la crida, aquesta es captura, permetent aprofitar el pas de dades entre el client que envia les comandes de versionat i el sistema FUSE.

Les metadades d'un fitxer s'emmagatzemen en el fitxer *.metadata* en el directori referent al fitxer que es versiona. El format per guardar les dades és en JSON<sup>6</sup>, i s'ha utilitzat la llibreria PARSON<sup>7</sup> que tracta amb aquest format, facilitant la seva gestió.

<sup>6</sup> "JSON.org." <http://www.json.org/>. Accessed 19 Oct. 2017.

<sup>7</sup> "kgabis/parson - GitHub." <https://github.com/kgabis/parson>. Accessed 19 Oct. 2017.

```

{
  "current": "1",
  "versions": [
    {
      "id": "0",
      "date": "",
      "tag": "DD MM AAAA MM:SS",
      "comment": "Comentari d'exemple"
    },
    {
      "id": "DD MM AAAA MM:SS",
      "date": "",
      "tag": "Tag d'exemple",
      "comment": "Comentari d'exemple"
    },
    ...
  ]
}

```

Figura 9. Estructura en format JSON del fitxer .metadata, emmagatzemat en cada directori referent a un fitxer automàtic que conté versions.



## Capítol 10. Anàlisi de Resultats

En aquest capítol s'analitzen els resultats obtinguts, respecte els objectius que es van proposar.

Les crides que permet el *Virtual Filesystem* per tal d'obtenir un sistema de fitxers funcional es troben implementades sense cap problema aparent a l'hora de gestionar el sistema de fitxers.

Pel que fa al versionat automàtic de fitxers, si consultem les dades que hi ha en el directori ocult referent a les versions d'un fitxer que hem anat modificant, trobarem les seves antigues versions amb el seu identificador com a nom del fitxer. A més, dins d'aquest també es troba el fitxer metadata amb les dades corresponents.

```
$ ls -l vefsdir/vefs.vers/exemple
total 16
-rw-rw-rw- 1 demian demian  5 oct 19 10:49 0
-rw-rw-rw- 1 demian demian 11 oct 19 10:49 1
-rw-rw-rw- 1 demian demian 18 oct 19 18:35 2
-rw-rw-rw- 1 demian demian 28 oct 19 18:36 3
-rw-rw-rw- 1 demian demian 228 oct 19 18:36 .metadata
```

Figura 10. Resultat que es mostra després de llistar el contingut del directori *exemple*. El directori *vefsdir* és el directori arrel del sistema de fitxers. El subdirectori *vefs.vers* és el directori ocult a l'usuari que conté els fitxers versionats. El fitxer *exemple* original es troba a *vefsdir*.

Existeixen alguns casos especials que s'han de tenir en compte. Per exemple, un programa pot generar fitxers temporals que el sistema tractarà de versionar igualment si es realitzen canvis sobre aquests. Aquest fet comporta que s'ompli l'espai dedicat a emmagatzemar les versions dels fitxers. En aquests casos, s'ha hagut d'analitzar un a un per veure de quina manera es podrien tractar.

Una de les funcionalitats ofertes a l'usuari és que el sistema mostri les versions disponibles d'un fitxer. Seguint l'exemple anterior, tenim el següent resultat:

```
$ ./versi show hola
Version 0:  date = "19 10 2017 10:49", tag = "", comment = "".
Version 1:  date = "19 10 2017 10:49", tag = "", comment = "".
Version 2:  date = "19 10 2017 18:35", tag = "", comment = "".
Version 3:  date = "19 10 2017 18:36", tag = "", comment = "".
```

Figura 10. Resultat que mostra la llista de versions disponibles d'un fitxer.

Una de les funcionalitats que no s'ha pogut enllestir a temps és la de restaurar la versió d'un fitxer. Tot i estar quasi a punt, el resultat que es veuria si es continués amb l'exemple anterior, restaurant la versió 2, és el següent:

```
$ ./versi restore hola
$ ./versi show hola
Version 0: date = "19 10 2017 10:49", tag = "", comment = "".
Version 1: date = "19 10 2017 10:49", tag = "", comment = "".
Version 2: date = "19 10 2017 18:35", tag = "", comment = "".
Version 3: date = "19 10 2017 18:36", tag = "", comment = "".
Version 4: date = "(Una data posterior)", tag = "", comment = "".
```

Figura 10. Resultat que mostra la llista de versions disponibles d'un fitxer després de restaurar la versió 2. La versió 4 conté l'últim estat en què es trobava el fitxer.

## Capítol 11. Conclusions

En aquest capítol es recorda els objectius del projecte que es van proposar per donar forma a la solució, quins han sigut assolits, i es menciona les desviacions que hagin pogut afectar al seu assoliment.

Aquests són els objectius que es van marcar per donar forma a la solució:

- 1) Ser un sistema de fitxers funcional.
- 2) Versionar fitxers de forma automàtica.
- 3) Permetre la gestió de versions.
- 4) Permetre realitzar còpies de seguretat.

D'entre aquests, s'han pogut assolir els dos primers, i en certa mesura el tercer. Degut a les desviacions durant la fase d'implementació per les dificultats per implementar algunes característiques, es va requerir més temps del previst que va ha deixat sense assolir l'últim objectiu. Per tant, es considera que el projecte soluciona parcialment el problema inicial plantejat.



## Justificació de les competències tècniques

En aquest apartat es justifiquen les competències tècniques assignades en l'elaboració d'aquest TFG. A continuació es llisten les competències que s'han assignat i es justifica com s'han desenvolupat.

### CTI3.1

Concebre sistemes, aplicacions i serveis basats en tecnologies de xarxa, tenint en compte Internet, web, comerç electrònic, multimèdia, serveis interactius i computació ubiqua. [En profunditat]

Justificació:

El projecte realitzat ha ajudat al desenvolupament de tècniques que permeten concebre sistemes basats en xarxa, ja que molts sistemes basats en tecnologies de xarxa requereixen d'un sistema de fitxers per funcionar. El sistema forma, per tant, una part del conjunt de tecnologies emprades en xarxa. Les competències adquirides en aquesta part, m'han ajudat a tenir més capacitat a l'hora de concebre sistemes basats en xarxa més eficients.

### CTI3.4

Dissenyar software de comunicacions. [Una mica]

Justificació:

El software dissenyat conté una part dedicada a comunicació entre client-servidor, tot i que no s'ha arribat a concebre en la implementació (objectiu 3)

## Referències

- [1] "Guide to OpenVMS File Applications - Software Products Library."  
[http://h30266.www3.hp.com/odl/i64os/opsys/vmsos84/documentation/pdf/OVMS\\_731\\_file\\_app.PDF](http://h30266.www3.hp.com/odl/i64os/opsys/vmsos84/documentation/pdf/OVMS_731_file_app.PDF).  
Accessed 28 Feb. 2017.
- [2] "Git - Git Basics." <https://git-scm.com/book/en/Getting-Started-Git-Basics>. Accessed 28 Feb. 2017.
- [3] "What Is ZFS? - Oracle Solaris ZFS Administration Guide."  
[https://docs.oracle.com/cd/E23823\\_01/html/819-5461/zfsover-2.html](https://docs.oracle.com/cd/E23823_01/html/819-5461/zfsover-2.html). Accessed 28 Feb. 2017.
- [4] Chapter 12 of "*Understanding the Linux Kernel*" by Daniel P. Bovet, Marco Cesati (O'Reilly Media, 2005), ISBN 0-596-00565-2
- [5] "Libfuse - GitHub." <https://github.com/libfuse/libfuse>. Accessed 19 Oct. 2017.
- [6] "JSON.org." <http://www.json.org/>. Accessed 19 Oct. 2017.
- [7] "kgabis/parson - GitHub." <https://github.com/kgabis/parson>. Accessed 19 Oct. 2017.